Comparing f_β-Optimal with Distance Based Merge Functions

Daan Van Britsom, Antoon Bronselaer, Guy De Tré

Department of Telecommunications and Information Processing, Ghent University
Sint-Pietersnieuwstraat 41, B-9000 Ghent, Belgium

Abstract. Merge functions informally combine information from a certain universe into a solution over that same universe. This typically results in a, preferably optimal, summarization. In previous research, merge functions over sets have been looked into extensively. A specific case concerns sets that allow elements to appear more than once, multisets. In this paper we compare two types of merge functions over multisets against each other. We examine both general properties as practical usability in a real world application.

Keywords: Merge functions, multisets, content selection

1 Introduction

In an ever growing digitalised world the amount of data available to the end user has very quickly become extremely cluttered. When one selects different data inputs regarding a single topic, frequently referred to as coreferent information, there is always duplicate, conflicting and missing information out and about. Therefore, when working with coreferent information there are several techniques that allow one to merge this information in order to get a briefer and correct overview. One of these possible techniques concerns the use of the f-value, a measurement balancing correctness and completeness, of the proposed solution with respect to the sources. These so-called f-optimal merge functions have been discussed extensively in [1] and expanded to f_β-optimal merge functions that allow for a preference to be given to either correctness or completeness by means of a parameter β. This type of merge function is typically applied to sets that allow elements to occurs multiple times, multisets. A second possible group of techniques concerns the use of distance measurements in order to determine which possible solution is closest related to all the sources. In order to illustrate how both types of merge functions can be useful in a real world application we demonstrate how their respective solutions can be used to generate multi-document summarizations (MDSs).

The remainder of this paper is structured as follows. In Section 2 we describe a few preliminary definitions required to understand the comparison we wish to establish in this document. Section 3 details how one is able to influence the outcome of an f_β-optimal merge function. A simple example of a distance based

merge function is provided in Section 4, whilst the comparison of both types of merge functions is made in Section 5. In part one of the latter we examine a few general properties and in part two we illustrate how both merge techniques can lie at the basis of Multi-Document Summarizations. Finally, we conclude in Section 6 with some final remarks on how we will further test the possibilities and advantages of both types of merge function in the creation of Multi-Document Summarizations.

2 Preliminaries

As a first type of merge function we would like to use in this paper's comparison, we iterate the definition of f_β-optimal merge functions. As stated in the introduction this type of merge functions is typically applied to sets, more specifically sets that allow elements to occur multiple times, multisets. We briefly recall some important definitions regarding multisets [2].

2.1 Multisets

Informally, a multiset is an unordered collection in which elements can occur multiple times. Many definitions have been proposed, but within the scope of this paper, we adopt the following functional definition of multisets [2].

Definition 1 (Multiset). *A multiset M over a universe U is defined by a function:*

$$M : U \to \mathbb{N}. \tag{1}$$

For each $u \in U$, $M(u)$ denotes the multiplicity of an element u in M. The set of all multisets drawn from a universe U is denoted $\mathcal{M}(U)$.

The j-cut of a multiset M is a regular set, denoted as M_j and given as:

$$M_j = \{u | u \in U \wedge M(u) \geq j\}. \tag{2}$$

Whenever we wish to assign an index $i \in \mathbb{N}$ to a multiset M, we use the notation $M_{(i)}$, while the notation M_j is preserved for the j-cut of M. We adopt the definitions of Yager [2] for the following operators: $\cup$, $\cap$, $\subseteq$ and $\in$.

2.2 Merge functions

The general framework of merge functions provides the following definition [3].

Definition 2 (Merge function). *A merge function over a universe U is defined by a function:*

$$\varpi : \mathcal{M}(U) \to U. \tag{3}$$

As explained in the introduction of this paper, we are interested in merge functions for (multi)sets rather than atomic elements. Therefore, we consider merge functions over a universe $\mathcal{M}(U)$ rather than a universe U. This provides us with the following function:

$$\varpi : \mathcal{M}\Big(\mathcal{M}(U)\Big) \rightarrow \mathcal{M}(U). \tag{4}$$

In order to avoid confusion, we shall denote S (a source) as a multiset over U and we shall denote M as a multiset over $\mathcal{M}(U)$ (a collection of sources). Thus, in general, M can be written as:

$$M = \big\{S_{(1)}, ..., S_{(n)}\big\}. \tag{5}$$

Finally, we shall denote $\mathscr{S} \in \mathcal{M}(U)$ as a general solution for a merge problem, i.e. $\varpi(M) = \mathscr{S}$. The most simple merge functions for multisets are of course the source intersection and the source union. That is, for any M:

$$\varpi_1(M) = \bigcap_{S \in M} S \tag{6}$$

$$\varpi_2(M) = \bigcup_{S \in M} S. \tag{7}$$

Within this paper, we consider a solution relevant if it is a superset of the source intersection or a subset of the source union. Therefore, we call the source intersection the lower solution (denoted $\underline{\mathscr{S}}$) and the source union the upper solution (denoted $\overline{\mathscr{S}}$). To conclude this section, we introduce the family of f-optimal merge functions, which are merge functions that maximize the harmonic mean of a measure of solution correctness (i.e. precision) and a measure of solution completeness (i.e. recall). This objective is better known as the f-value [4]. To adapt the notion of precision and recall to the setting of multiset merging, we define two *local* (i.e. element-based) measures [1].

Definition 3 (Local precision and recall). *Consider a multiset of sources $M = \big\{S_{(1)}, ..., S_{(n)}\big\}$. Local precision and recall are defined by functions p^* and r^* such that:*

$$\forall u \in U : \forall j \in \mathbb{N} : p^*(u, j|M) = \frac{1}{|M|} \sum_{S \in M \wedge S(u) \geq j} M(S) \tag{8}$$

$$\forall u \in U : \forall j \in \mathbb{N} : r^*(u, j|M) = \frac{1}{|M|} \sum_{S \in M \wedge S(u) \leq j} M(S). \tag{9}$$

One can see that p^* depicts the percentage of sources in which u occurs at least j times and r^* the percentage of sources in which u occurs a maximum of j times.

Definition 4 (f-optimal merge function). *A merge function ϖ is f-optimal if it satisfies for any $M \in \mathcal{M}(\mathcal{M}(U))$:*

$$\varpi(M) = \underset{\mathscr{S} \in \mathcal{M}(U)}{\arg\max}\, f(\mathscr{S}|M) = \underset{\mathscr{S} \in \mathcal{M}(U)}{\arg\max} \left(\frac{2 \cdot p(\mathscr{S}|M) \cdot r(\mathscr{S}|M)}{p(\mathscr{S}|M) + r(\mathscr{S}|M)} \right) \tag{10}$$

constrained by:

$$\left(\max_{\mathscr{S}\in\mathcal{M}(U)} f(\mathscr{S}|M) = 0\right) \Rightarrow \varpi(M) = \emptyset \tag{11}$$

and where, with T *a triangular norm, we have that:*

$$p(\mathscr{S}|M) = \mathop{\mathrm{T}}_{u\in\overline{\mathscr{S}}}\left(p^*(u, \mathscr{S}(u)|M)\right) \tag{12}$$

$$r(\mathscr{S}|M) = \mathop{\mathrm{T}}_{u\in\overline{\mathscr{S}}}\left(r^*(u, \mathscr{S}(u)|M)\right). \tag{13}$$

3 Influencing the content selection

The f-optimal merge function as defined in Definition 4 doesn't allow one to influence the outcome $\mathscr{S} \in \mathcal{M}(U)$ of the merge function. Suppose one would want to select fewer elements in order to show a preference to precision rather than recall. In order to do so one could take a subset of $\mathscr{S}$ but then one would no longer have a solution with an optimal f-value. The merge function becomes even more restricting if one would want more elements as a solution, thus giving preference to recall rather than precision, for there is no option to gain more concepts. In order to influence the outcome of the f-optimal merge function we have chosen to use the weighted harmonic mean [5], and the merge function thus changes as follows.

Definition 5 (Weighted f_β-optimal merge func.). *A merge function ϖ is f_β-optimal if it satisfies for any $M \in \mathcal{M}(\mathcal{M}(U))$:*

$$\varpi(M) = \operatorname*{arg\,max}_{\mathscr{S}\in\mathcal{M}(U)} f_\beta(\mathscr{S}|M) = \operatorname*{arg\,max}_{\mathscr{S}\in\mathcal{M}(U)} \left(\frac{(1+\beta^2)\cdot p(\mathscr{S}|M)\cdot r(\mathscr{S}|M)}{\beta^2\cdot p(\mathscr{S}|M) + r(\mathscr{S}|M)}\right) \tag{14}$$

still constrained by (11), $\beta \in [0, \infty]$ and where, with T *a triangular norm, (12) and (13) still apply.*

The parameter β expresses how much more weight is given to recall as opposed to precision, more specifically, recall has a weight of β times precision. Thus, when $\beta = 1$ precision and recall are weighted the same and this results in the non-weighted f-optimal merge function as defined in Definition 4. When $\beta < 1$, a preference is given to precision, for example when $\beta = 0.5$, recall is given half the weight of precision. When $\beta > 1$, a preference is given to recall, for example when $\beta = 2$, recall is given twice the weight of precision. When $\beta = 0$, f_β returns the precision and when β approaches infinity f_β results in the recall.

In previous research it has been shown that the specific case where $T = \mathrm{T}_\mathbf{M}$, the minimum t-norm as proposed by Zadeh, has interesting properties and therefore, for the remainder of this paper, we will restrict ourselves to this case [1].

4 Distance based merge functions

Another approach to generate a result for a set of coreferent items one wishes to merge consists of using distance based merge functions. There are quite a few techniques to measure a distance between two sets, including Cosine similarity and Minkowski distances such as the Manhattan and Euclidean distance. The example we will be using throughout this paper is based on the Minkowski distance, an effective and frequently used distance measurement.

Definition 6 (Simple distance function). *Consider two sources $S_{(1)}, S_{(2)}$ and a universe U consisting of u elements. The distance between these sources according to the simple distance function δ is*

$$\delta(S_{(1)}, S_{(2)}) = \sum_{u \in U} |S_{(1)}(u) - S_{(2)}(u)| \tag{15}$$

with, as stated earlier, $S_{(i)}(u)$ the multiplicity of element u in source $S_{(i)}$

This distance function results in calculating the number of adjustments required to get from one source to another, whilst only allowing additive and subtractive operations.

One can now use the distance function δ to calculate the distance from a single set with respect to several different sets.

Definition 7 (Simple distance based merge function). *Consider a multiset of sources $M = \{S_{(1)}, ..., S_{(n)}\}$ in a universe U. A distance based merge function ϖ_δ returns the solution that has a minimal total distance to all provided sources. For each element $m \in M$ for each $M \in \mathcal{M}(\mathcal{M}(U))$:*

$$\varpi_\delta(M) = \arg\min \sum_{i=1}^{n} \delta(\mathscr{S}, S_{(i)}) \tag{16}$$

Informally, the function ϖ_δ calculates the solution $\mathscr{S}$ that requires the least total additive and subtractive operations to go from $\mathscr{S}$ to all the possible sources S in M.

Due to the distributivity of the minimum over the summation we can formulate this distance function as follows.

Definition 8 (Element based merge func.). *Consider a multiset of sources $M = \{S_{(1)}, ..., S_{(n)}\}$ in a universe U. A distance based merge function $\varpi_{\delta\epsilon}$ returns the solution that has a minimal total distance to all provided sources. For each element $m \in M$ for each $M \in \mathcal{M}(\mathcal{M}(U))$:*

$$\varpi_{\delta\epsilon}(M) = \forall u \in U : \mathscr{S}(u) = \arg\min_{k \in \mathbb{N}} \sum_{i=1}^{n} |S_{(i)}(u) - k| \tag{17}$$

Informally, the function $\varpi_{\delta\epsilon}$ calculates the optimal multiplicity (range of multiplicities) $\mathscr{S}(u)$ for each element u so that requires the least total additive and subtractive operations to go from $\mathscr{S}(u)$ to the multiplicity of that element in every source $S_{(i)}$ in M.

Obviously, the complexity of the latter function is a lot smaller than the complexity of ϖ_{δ}. However, it quickly becomes apparent that if we were to apply this function on a realistic dataset of documents we would have an exponential amount of possible solutions to compare. If the dataset only consists of a universe of 100 words with a average multiplicity range of only five possibilities, we would have to generate and evaluate 5^{100}, roughly $7.8 * 10^{69}$ solutions. The solution space is however uniquely defined by the multiplicityset generated by $\varpi_{\delta\epsilon}$.

5 Making the Comparison

Now that both types of merge functions have been recapitulated we want to compare them to one another. In subsection 5.1 we go over a few useful properties concerning merge functions and see which ones apply on either one of the types of function. In subsection 5.2 we apply both functions to a real world application, the summarizing of multiple documents, more specifically the content selection step, and see which advantages or disadvantages the merge functions have.

5.1 Properties

Property 1 (Idempotence). A merge function ϖ for multisets over a ground universe U is idempotent if and only if, for any $M = \{S, ..., S\}$ we have that:

$$\varpi(M) = S. \tag{18}$$

As has been proven in [1], the f-optimal merge function is idempotent, the proof that the weighted f_β-optimal merge function is idempotent as well is trivial. It is obvious that the proposed distance based merge function is idempotent as well, considering that the solution S is the only one not requiring any additive or subtractive operations relative to all the sources.

Property 2 (Monotonicity). A merge function ϖ for multisets over a ground universe U is monotone if and only if, for any $M = \{S_{(1)}, ..., S_{(n)}\}$ and for any $M^* = \{S^*_{(1)}, ..., S^*_{(n)}\}$ such that:

$$\forall i \in \{1, ..., n\} : S_{(i)} \subseteq S^*_{(i)} \wedge M\left(S_{(i)}\right) = M^*\left(S^*_{(i)}\right) \tag{19}$$

we have that:

$$\varpi(M) \subseteq \varpi(M^*). \tag{20}$$

Where the defined global precision and recall functions are monotone as proven in [1], the f-optimal merge function is not and thus the weighted f_β-optimal merge function is neither. Due to the nature of the Minkowski distance, the proposed distance based merge function however, is monotone.

Property 3 (Quasi Robustness). A merge function ϖ over $\mathcal{M}(U)$ is quasi-robust if and only if, for any error-free $M \in \mathcal{M}(\mathcal{M}(U))$ (with $|M| > 1$) and for any erroneous source E, we have that:

$$\varpi(M \cup \{E\}) \cap E = \emptyset. \tag{21}$$

With E an erroneous source, as defined in [6], a source that has no element in common with any of the sources in M.

It has been proven in [6] that the f-optimal merge function is quasi robust. The f_β-optimal merge function however is not. When β approaches infinity the f_β-optimal merge functions approaches the union for which quasi robustness clearly doesn't hold. The proposed distance based merge function however is quasi robust as well from the moment that $|M| > 2$. The proof for this is trivial because the moment you have two sources not containing a certain element, including this element to the solution will always result in at least one more additive or subtractive operation relative to the sources as opposed to not including it into the solution.

5.2 Multi-Document Summarization

In order to illustrate other possible differences between distance based and f_β-optimal merge function we apply both algorithms to the Multi-Document Summarization problem (MDS problem) using the Document Understand Conference dataset of 2002 (DUC2002) and try to evaluate how we can influence both algorithms. Suppose we therefore define a cluster of sources from the DUC2002 set as a multiset M and every document of the n documents of that cluster as a source S so the equation $M = \{S_{(1)}, ..., S_{(n)}\}$ clearly still holds up. The solution $\mathscr{S}$ of the merge function can only contain elements from the sources, therefore the universe U does not consist of the entire English language but instead contains all the words from all the different sources $\{S_{(1)}, ..., S_{(n)}\}$ that are part of the cluster cluster combined.

It has been shown in previous research that once a set of key concepts has been identified for a cluster of coreferent documents, a summarization can be generated [7]. In this paper we will therefore focus on how both types of merge functions can generate a set of concepts that represent the key elements of the cluster automatically and as usable as possible. We will focus on two separate issues. First, we will try to establish how easy it is to find a single optimal set of key concepts defining the cluster. Secondly, we will examine to which extend it is possible to objectively influence this selection process.

f_β-optimal merge function If we were to illustrate the type of solution generated by the f_β-optimal merge function by using the first cluster of documents of the DUC2002 set we would get, for a value of β of 1, thus resulting in the non-weighted f-optimal merge function, the following result.

$\varpi_{\beta=1}(M)$ = {{*weather=1, winds=5, rico=3 ...*
..., director=1, inches=1, service=1} =1, {*caribbean=2, like=1, residents=1 ...*
..., civil=1, expected=1, only=1}=1}

As one can see above, for the first cluster we get a multiset containing two other multisets with multiplicity one as a solution. When we calculate the solution for each cluster of the DUC2002 set we get a small multiset as a result each time, as one can see in Table 1. The distance based merge function however, as one can read further down in the paper, does not. This makes it a lot more difficult to choose one of the suggested multisets and later on influence this multiset.

ID	$\sharp\mathscr{S}$	ID	$\sharp\mathscr{S}$	ID	$\sharp\mathscr{S}$	ID	$\sharp\mathscr{S}$	ID	$\sharp\mathscr{S}$	ID	$\sharp\mathscr{S}$
1	2	11	2	21	2	31	2	41	2	51	2
2	1	12	1	22	2	32	1	42	2	52	1
3	2	13	1	23	2	33	1	43	1	53	1
4	1	14	2	24	1	34	2	44	2	54	1
5	2	15	2	25	1	35	2	45	2	55	2
6	1	16	2	26	1	36	2	46	1	56	2
7	2	17	2	27	2	37	2	47	2	57	2
8	1	18	2	28	2	38	1	48	2	58	1
9	2	19	1	29	2	39	1	49	1	59	2
10	1	20	1	30	2	40	2	50	2		

Table 1: Number of solutions per clusterID for the DUC2002 dataset for the f_β-optimal function with $\beta = 1$

The next evaluation step concerns testing the amenability of the f_β-optimal merge function. As has been recollected in Section 3 this can be done by the usage of the parameter β. We illustrate again by using the first cluster of the DUC2002 dataset.

$\varpi_{\beta=0.25}(M)$ = {{*to=3, gilbert=2, storm=3, caribbean=1, mph=1, were=1, west=1, national=1, in=6, said=3, was=2, the=23, on=2, winds=1, s=3, hurricane=6, at=1, they=1, of=10, from=1, moving=1, for=1, center=1, a=4, coast=1, and=10*}=1}

$\varpi_{\beta=0.50}(M)$ = {{ *national=1, center=2, puerto=1, gilbert=5, flooding=1, we=1, this=1, at=3, sustained=1, as=1, caribbean=1, would=1, moving=1, one=1, an=1, residents=1, 000=1, islands=1, weather=1, from=3, hurricane=6, they=1, into=1, was=4, miami=1, republic=1, west=1, about=2, people=1, dominican=1, coast=1, inches=1, it=3, is=1, the=30, in=10, on=2, said=5, of=12, mph=2, with=2, by=1, for=3, s=3, their=1, off=2, and=10, were=1, night=1, storm=4, reported=1, winds=4, to=6, a=5, sunday=1, heavy=1, there=1*}=1}

For values of $\beta < 1$ one obtains a subset of the original solution obtained from the non-weighted f-optimal merge function, as proven in [8]. As one can see above this may also result in the fact that the solution $\mathscr{S}$ no longer contains several multisets. The reason for this can be found in the fact that a preference is given to precision, to correctness, and therefore the likelihood of multiple multisets providing an equally optimal solution, drops. The same conclusion can

be made as when $\beta > 1$, due to the fact that a preference is given to recall, the likelihood of multiple multisets being part of the optimal solution drops, as can be seen in the example.

Simple distance based merge function As we have shown in Section 4 it might prove to be difficult to generate and display all possible results. But, as previously stated, the solution space is uniquely defined by the multiplicityset generated as described in Definition 8. We once more illustrate the results of this type of merge function by generating a solution for the first cluster of the DUC2002 set. The multiplicityset defining all the possible solutions for the first cluster can be found in Appendix A. Suffice to say it contains over 100 words, some of which with over 5 possible optimal multiplicities, which makes it very impractical to use due to the large amount of possible solutions.

Why there are so many possible solutions lies in the fact the more documents we have in which a word occurs, the higher the chance that there is not a single multiplicity defining the optimal balance. For instance, if a word u were to occur one time in the first source $S_{(1)}(u) = 1$, three times in the second $S_{(2)}(u) = 3$, $S_{(3)}(u) = 5$ and $S_{(4)}(u) = 7$, then the solution $\mathscr{S}(u)$ exists out of three possible multiplicities $\mathscr{S}(u) = [3, 4, 5]$ because from each multiplicity it only requires a total of eight additions or subtractions relative to the occurrences in the sources.

The reason why we still care about this difference is due to the fact that the semantic difference between a word w having multiplicity one or zero makes a huge difference in the interpretation by the user but for the distance function it makes virtually no difference at all. That is why there are so many possible optimal solutions. This of course only occurs when the sources are rather well balanced. It is also perfectly possible that there is only a single correct multiplicity for every word. However as one can clearly see in Table 2 depicting the amount of possible solutions for each cluster, as soon as there is not a single optimal solution the amount of possible solutions runs extremely high. This of course makes it very difficult to choose an optimal solution and afterwards influence the content selection.

ID	$\sharp\mathscr{S}$	ID	$\sharp\mathscr{S}$	ID	$\sharp\mathscr{S}$	ID	$\sharp\mathscr{S}$	ID	$\sharp\mathscr{S}$	ID	$\sharp\mathscr{S}$
1	3.377E44	11	5.107E18	21	1.297E18	31	5.629E16	41	3.486E21	51	3.799E15
2	1	12	1	22	6.333E15	32	1	42	2.111E14	52	1
3	1.480E31	13	1	23	3.239E54	33	1	43	1	53	1
4	1	14	6.984E40	24	1	34	1.367E17	44	1.159E11	54	1
5	1.776E36	15	6.648E19	25	1	35	1.669E13	45	3.298E13	55	1.489E31
6	1	16	1.290E25	26	1	36	1.202E16	46	1	56	1.513E18
7	8.881E35	17	2.988E22	27	4.669E19	37	2.350E27	47	1.056E14	57	2.757E21
8	1	18	7.124E15	28	4.178E30	38	1	48	8.977E16	58	1
9	9.277E11	19	1	29	3.804E15	39	1	49	1	59	6.274E26
10	1	20	1	30	1.197E36	40	1.284E32	50	1.811E9		

Table 2: Number of solutions per clusterID for the DUC2002 dataset for the distance based merge function

In order to illustrate how the proposed distance based merge function would generate a multiset of keywords κ of a set of documents, we select a few of the possible multisets of keywords with a minimal total distance to all the sources.

- κ_{min} generated by using the smallest multiplicity per element
- κ_{max} generated by using the largest multiplicity
- κ_{med} generated by using the median multiplicity of each element

One can find κ_{min} and κ_{max} completely in Appendix B. As one would suspect κ_{med} generated by using the median multiplicity of each element is analogue to κ_{max} with maximum multiplicity however it might introduce certain difference for elements that are on the cusp, for instance multiplicity range one to zero. It is therefore not present in Appendix B.

Practically speaking, besides the issue that there is an enormous amount of possible sets of key concepts, it is also quite difficult to objectively influence this selection process. One of the great advantages of the f_β-optimal merge function lies in the fact that through changing the parameter β one can influence the outcome of the function. When applying the merge function $\varpi_{\delta\epsilon}$ one frequently has an extreme amount of possible optimal solutions to select a set of concepts from. One might see the choice herein as possibly influencing the outcome, but one might lose valuable information just because other words appear in the same average frequency and get lost in the selection process. An objective way to influence the selection process would be to use another distance function but unless we find a more efficient technique to calculate the merge function the performance and usability of this merge function will be extremely poor.

6 Conclusion and Future Work

In this paper we have made a comparison between a weighted f_β-optimal merge function and a simple distance based merge function. We compared a few general properties concerning merge functions that showed that both functions have their merit, but when it came down to usability in a real life problem the f_β-optimal merge function proved to be performing better. The f_β-optimal merge function however has been developed more and is more advanced than the proposed distance based merge function. As previously stated there are several other distance functions we could apply in order to calculate the distance between two sources. Other possibilities include, but one is not restricted to, the Cosine similarity, Hamming distance and other variances on the Minkowksi distance. We are planning to investigate these further but the initial research concerning these measurements falls outside of the scope of this paper.

References

1. Antoon Bronselaer, Daan Van Britsom, Guy De Tré: A framework for multiset merging. Fuzzy Sets and Systems **191**(0) (2012) 1 – 20

2. Ronald Yager: On the theory of bags. International Journal of General Systems **13**(1) (1986) 23–27
3. Antoon Bronselaer, Guy De Tré, Daan Van Britsom: Multiset merging: the majority rule. In: Proceedings of the EUROFUSE 2011 Workshop. (2011)
4. Ricardo Baeza-Yates, Berthier Ribeiro-Neto: Modern information retrieval. ACM Press (1999)
5. C. J. van Rijsbergen: Information Retrieval. Butterworths, London (1979)
6. Antoon Bronselaer, Daan Van Britsom, Guy De Tré: Robustness of multiset merge functions. In: Proceedings of the 14th International conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Springer (2012)
7. Daan Van Britsom, Antoon Bronselaer, Guy De Tré: Automatically generating multi-document summarizations. In: Proceedings of the 11th International conference on intelligent systems design and applications, IEEE (2011) 142–147
8. Daan Van Britsom, Antoon Bronselaer, Guy De Tré: Concept identification in constructing multi-document summarizations. In: Proceedings of the 14th International conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Springer (2012)

Appendices

A MultiplicitySet generated by $\varpi_{\delta\epsilon}$

MultiplicitySet = {*time:[1], right:[0, 1], 3:[1], 2:[0, 1], 5:[0, 1], lines:[0, 1], a:[6], m:[1], s:[5, 6], p:[0, 1], zone:[0, 1], bob:[0, 1], hal:[0, 1], strengthened:[0, 1], had:[1, 2], watch:[1], areas:[0, 1], reached:[0, 1], 000:[1, 2], moved:[1], expected:[1], which:[0, 1], there:[1, 2], reported:[1, 2], puerto:[3], western:[0, 1], hurricanes:[0, 1], home:[0, 1], television:[0, 1], tropical:[1], officials:[1], gerrish:[0, 1], cut:[0, 1], jamaica:[3, 4, 5], where:[0, 1], hit:[1], eye:[0, 1, 2], damage:[0, 1], strong:[0, 1], streets:[0, 1], gilbert:[5], while:[0, 1], east:[1], into:[1], night:[2], along:[1], miami:[1], sunday:[1, 2], caribbean:[1, 2], seen:[0, 1], south:[2, 3], down:[0, 1], province:[0, 1], islands:[1, 2], hurricane:[10, 11, 12, 13], strength:[0, 1], ripped:[0, 1], high:[1], people:[1, 2], arrived:[0, 1], slammed:[0, 1], like:[0, 1], coastal:[1], now:[0, 1], residents:[1], radio:[0, 1], but:[1], saturday:[1], north:[1, 2], southeast:[0, 1, 2], haiti:[0, 1, 2], around:[1], sheets:[1], their:[1], first:[0, 1], said:[10], higher:[0, 1], storm:[4, 5, 6], over:[1, 2], government:[0, 1], moving:[1, 2], he:[0, 1], miles:[3], before:[1], ocean:[0, 1], sustained:[1], warnings:[1, 2], by:[1, 2], long:[1], kingston:[0, 1], would:[2], be:[0, 1], get:[0, 1], and:[18, 19, 20], maximum:[0, 1], island:[2, 3], area:[0, 1], edt:[0, 1], formed:[1], all:[1], at:[4], dominican:[2, 3], as:[4, 5], an:[1, 2], off:[2, 3], forecaster:[1], they:[2], no:[1, 2], of:[19, 20, 21, 22, 23], on:[4, 5], only:[1], or:[0, 1], winds:[5, 6], most:[1], flights:[0, 1, 2], larger:[0, 1], second:[0, 1], gulf:[1], when:[0, 1], certainly:[0, 1], republic:[2, 3], issued:[1], heavy:[2, 3], eastern:[0, 1], this:[1, 2], from:[3, 4, 5], was:[4, 5], is:[1, 2], it:[5, 6, 7], know:[0, 1], in:[12], hotel:[0, 1], mph:[3, 4], passed:[0, 1], westward:[0, 1], forecasters:[0, 1, 2], cayman:[0, 1, 2], windows:[0, 1], 25:[0, 1], we:[1, 2, 3], next:[0, 1], 15:[0, 1], northwest:[1], ve:[0, 1], civil:[0, 1], up:[0, 1], 10:[1], to:[10, 11, 12], reports:[0, 1], mexico:[1], that:[2, 3, 4, 5, 6], about:[2], re:[0, 1], rain:[1, 2], defense:[0, 1], track:[0, 1], inches:[1], service:[1],*

our:[0, 1], out:[0, 1], 50:[0, 1], flooding:[1], flash:[0, 1], for:[4, 5], city:[1, 2], center:[3], weather:[1], national:[2, 3], director:[1], trees:[1], cuba:[0, 1, 2], evacuated:[0, 1], southern:[0, 1], 100:[1, 2], should:[1], canceled:[0, 1], little:[0, 1], were:[4, 5, 6, 7], three:[0, 1], power:[1], systems:[1], west:[2], other:[0, 1], one:[1, 2], coast:[3, 4], rico:[3], with:[3, 4], the:[40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56], roofs:[1], continue:[0, 1]}

B Complete mergesets generated by $\varpi_{\delta\epsilon}$

κ_{min} = {*weather=1, winds=5, rico=3, their=1, power=1, puerto=3, hit=1, most=1, island=2, hurricane=10, issued=1, this=1, one=1, northwest=1, sustained=1, expected=1, islands=1, we=1, high=1, mexico=1, dominican=2, for=4, south=2, reported=1, about=2, systems=1, heavy=2, over=1, north=1, warnings=1, republic=2, sunday=1, only=1, night=2, jamaica=3, rain=1, but=1, east=1, it=5, is=1, tropical=1, caribbean=1, in=12, sheets=1, before=1, residents=1, s=5, said=10, on=4, coastal=1, that=2, 100=1, off=2, m=1, with=3, 000=1, of=19, by=1, had=1, moving=1, around=1, a=6, from=3, time=1, should=1, national=2, no=1, and=18, to=10, formed=1, center=3, at=4, there=1, as=4, along=1, west=2, an=1, flooding=1, forecaster=1, 3=1, moved=1, they=2, would=2, people=1, officials=1, roofs=1, 10=1, storm=4, saturday=1, miles=3, city=1, mph=3, watch=1, all=1, gilbert=5, into=1, were=4, miami=1, was=4, coast=3, the=40, long=1, trees=1, gulf=1, director=1, inches=1, service=1*}

κ_{max} = {*caribbean=2, like=1, residents=1, that=6, seen=1, puerto=3, 100=2, damage=1, officials=1, warnings=2, inches=1, where=1, into=1, get=1, higher=1, sheets=1, trees=1, we=3, watch=1, western=1, jamaica=5, coast=4, national=3, hurricane=13, southern=1, service=1, around=1, mph=4, radio=1, reached=1, edt=1, ve=1, maximum=1, it=7, reports=1, is=2, hotel=1, in=12, up=1, which=1, evacuated=1, down=1, hit=1, the=56, was=5, gerrish=1, larger=1, certainly=1, city=2, arrived=1, little=1, heavy=3, track=1, he=1, one=2, to=12, center=3, but=1, north=2, first=1, defense=1, three=1, along=1, when=1, this=2, westward=1, south=3, next=1, sunday=2, republic=3, people=2, power=1, other=1, passed=1, right=1, and=20, eastern=1, high=1, islands=2, island=3, most=1, over=2, re=1, while=1, eye=2, gilbert=5, canceled=1, slammed=1, rain=2, miami=1, issued=1, 000=2, area=1, miles=3, haiti=2, night=2, ripped=1, 50=1, tropical=1, all=1, windows=1, time=1, ocean=1, about=2, television=1, their=1, flights=2, flooding=1, strength=1, strengthened=1, southeast=2, with=4, flash=1, storm=6, director=1, they=2, now=1, cuba=2, s=6, p=1, out=1, m=1, weather=1, long=1, our=1, or=1, systems=1, moving=2, on=5, kingston=1, cayman=2, coastal=1, gulf=1, a=6, of=23, formed=1, by=2, west=2, zone=1, dominican=3, said=10, areas=1, for=5, from=5, should=1, winds=6, moved=1, be=1, no=2, hurricanes=1, reported=2, 25=1, lines=1, cut=1, roofs=1, at=4, as=5, mexico=1, 5=1, an=2, before=1, bob=1, 3=1, 2=1, were=7, know=1, saturday=1, forecaster=1, east=1, streets=1, 15=1, sustained=1, 10=1, there=2, hal=1, province=1, would=2, government=1, second=1, home=1, had=2, rico=3, strong=1, northwest=1, continue=1, off=3, civil=1, forecasters=2, expected=1, only=1*}